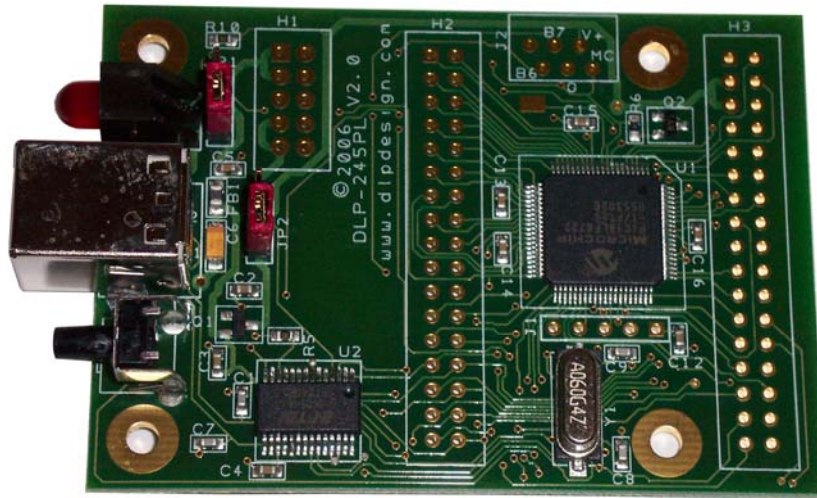




## DLP-245PL-G

### USB / Microcontroller Module (Lead-Free)



The DLP-245PL-G combines the same lead-free USB interface used in the DLP-USB245M-G module with a Microchip PIC microcontroller to form a rapid development tool. The 18LF8722 microcontroller is preprogrammed with basic functionality for accessing the port pins and can be reprogrammed with user hex code via a 5-pin header that is compatible with Microchip's MPLAB ICD2 device programmer/debugger (purchased separately).

#### **DLP-245PL FEATURES**

- Send/receive data over USB to a host computer at up to 2 megabits per second
- 50 digital I/O lines (16 can be configured as A/D inputs) plus the 8-bit data bus available for interfacing to user electronics
- Microchip 18LF8722 processor with 128K FLASH Program ROM, 3840 bytes RAM, 1024 bytes of data EEPROM and a multi-channel, 10-bit A/D converter
- "Token I/O" code preprogrammed into the 18LF8722's FLASH for basic port pin input/output capability including access to the A/D, EEPROM, and external digital temperature sensors. ROM can be erased and reprogrammed with a device programmer (purchased separately).
- No in-depth knowledge of USB is required as all USB protocol is handled automatically by the on-board FT245RL and its support circuitry
- Royalty-free device drivers eliminate the need for USB driver development in most cases

- Required 5V supply can be taken directly from the USB port or supplied by user electronics
- USB 1.1 and USB 2.0 compatible
- USB VID, PID, serial number, and product description strings stored in on-chip EEPROM
- Virtual Com Port (VCP) drivers for:
  - Windows 98 and Windows 98 SE
  - Windows 2000/ME/XP
  - Windows CE\*\*
  - MAC OS-8 and OS-9
  - MAC OS-X
  - Linux 2.40 and greater
- D2XX (USB direct drivers + DLL S/W interface)
  - Windows 98 and Windows 98 SE
  - Windows 2000/ME/XP

## APPLICATION AREAS

- Prototype development
- USB ISDN and ADSL modems
- USB interface for digital cameras
- USB interface for MP3 players
- High-speed USB instrumentation
- USB smart-card readers
- Set Top Box (STB) PC-USB interface
- USB hardware modems
- USB wireless modems
- USB bar code readers

## GENERAL DESCRIPTION

The DLP-245PL-G provides a cost-effective, microcontroller-based method of interfacing an electronic peripheral to a host computer via USB.

To send data from the peripheral to the host computer, the microcontroller simply writes the byte-wide data into the FT245RL when TXE# is low. If the FT245RL's transmit buffer fills up or is busy storing the previously written byte, it will take its TXE# high in order to stop further data from being written until some of the FIFO data has been transferred over USB to the host.

When the host sends data to the peripheral over USB, the FT245RL will take RXF# low to let the microcontroller know that at least one byte of data is available. The microcontroller then reads the data until RXF# goes high indicating that no more data is available to read.

By using FTDI's Virtual COM Port drivers, the peripheral looks like a standard COM port to the application software. Commands to set the baud rate are ignored--the FT245RL always transfers data at its fastest rate regardless of the application's baud-rate setting. The latest versions of the drivers are available for download from DLP Design's website at [www.dlpdesign.com](http://www.dlpdesign.com).

## DRIVER SOFTWARE

FTDI's VCP (Virtual COM Port) driver files are provided royalty free on the condition that they are used only with designs incorporating an FTDI device (i.e. the FT245RL and DLP-245PL-G). The latest version of the drivers can be downloaded from [dlpdesign.com](http://dlpdesign.com) or [ftdichip.com](http://ftdichip.com).

The VCP driver download file is a combined set of drivers for Windows 98, Windows ME, and Windows 2000/XP. Unzip the file to a blank floppy disk or folder on your PC. (The drivers can coexist on the same floppy disk or folder since the INF files determine which set of drivers to load for each operating system version.) Once loaded, the VCP drivers allow your application software running on the host PC to communicate with the DLP-245PL-G as though it were connected to a COM (RS-232) port.

In addition to VCP drivers, FTDI's D2XX direct drivers for Windows offer an alternative solution to the VCP drivers that allow application software to interface with the DLP-245PL-G using a DLL instead of a Virtual Com Port. The architecture of the D2XX drivers consists of a Windows WDM driver that communicates with the FT245RL device via the Windows USB stack and a DLL that interfaces the application software (written in VC++, C++ Builder, Delphi, VB, etc.) to the WDM driver. An INF installation file, uninstaller program, and D2XX Programmer's Guide complete the package.

The D2XX direct drivers add support for simultaneous access and control of multiple FT245RL devices. The extended open function (FT\_OpenEx) allows the device to be opened either by its product description or serial number, both of which can be programmed to be unique. The list devices function (FT\_ListDevices) allows the application software to determine which devices are currently available for use, again by product description or by serial number.

Functions are provided to program the EEPROM (FT\_EE\_Program) and read the EEPROM (FT\_EE\_Read). Unused space in the EEPROM is called the user area (EEUA), and functions are provided to access the EEUA. FT\_EE\_UASize gets its size, FT\_EE\_UAWrite writes data into it, and FT\_EE\_UARead is used to read its contents.

***Download FTDI Application Notes AN232-03, AN232-05, AN232-06, and AN232-07 for detailed instructions on how to install and remove the drivers.***

## EEPROM WRITE UTILITY

The DLP-245PL-G has the option to accept manufacturer-specific information that is written into the on-chip EEPROM. Parameters that can be programmed include the VID and the PID identifiers, the manufacturer's product string, and a serial number.

MPROG is the latest EEPROM serializer utility from FTDI for the FT245RL device. MPROG is based on the new D2XX drivers and will work on Windows 98, Windows ME, and Windows 2000/XP platforms. You must install the latest release of the D2XX drivers in order to run this application.

If you have VCP drivers installed on the PC that will be used to perform the EEPROM write process, you must uninstall these drivers and install the D2XX drivers prior to running MPROG.

## QUICK START GUIDE

***This guide requires the use of a Windows 98/2000/XP PC that is equipped with a USB port.***

1. Download the DLL version of the device drivers from either **dlpdesign.com** or **ftdichip.com**. Unzip the drivers onto a blank floppy disk or into a folder on the hard drive.
2. The DLP-245PL-G can be configured to receive its operating power from the USB port or from an external power source. Jumper JP1 allows for this configuration. Select JP1 Pins 1 and 2 if the DLP-245PL-G is to take its power from the USB port on the host computer; jumper JP1 Pins 2 and 3 if the DLP-245PL-G is to take its power from your external power source.
3. Jumper JP2 selects the power source for the 18LF8722 microcontroller and the interface pins on the FT245RL USB chip. Select JP2 Pins 1 and 2 if 5 volts from the USB port is to be used to power the microcontroller. Select JP2 Pins 2 and 3 if an external supply is to be used to power the microcontroller.

Note: The board will not operate until a power source has been selected as mentioned in Steps 2 and 3.

4. Connect the DLP-245PL-G board to the PC via a standard A-B, 6-foot USB cable. This action initiates the loading of the USB drivers. When prompted, select the folder where the DLL version of the device drivers was stored in Step 1. Windows will then complete the installation of the device drivers for the DLP-245PL-G board. The next time the DLP-245PL-G board is attached, the host PC will immediately load the correct drivers without any prompting. Reboot the PC if prompted to do so.

The DLP-245PL-G is shipped with default VID, PID, etc. values programmed into the EEPROM. You only need to run MPROG if you want to change the default values.

At this point, the DLP-245PL-G is ready for use. Note that the DLP-245PL-G will appear non-responsive if data sent from the host PC is not read from the FT245RL device by the 18LF8722 microcontroller.

If changing drivers from the VCP to the DLL type (or vice versa), you must first uninstall the currently loaded drivers. This is accomplished by first disconnecting the DLP-245PL-G adapter from the host computer and then running Add-Remove-Programs (via Control Panel) to remove the currently loaded drivers.

## TOKEN I/O

The 18LF8722 microcontroller on the DLP-245PL-G is shipped from the factory preprogrammed with firmware that provides rudimentary access to the port pins via either the VCP or DLL drivers. Features include the ability to read and write individual port pins as well as 8-bit port reads and writes.

The firmware in the DLP-245PL-G also provides access to the 18LF8722's A/D converter, EEPROM memory, and communications with digital temperature-sensing devices.

Commands sent to the Token I/O firmware must adhere to a specific communications protocol. Each command sequence contains the following information:

<b>Byte 0:</b>	Number of bytes in command sequence
<b>Byte 1:</b>	Command
<b>Byte 2...n-1:</b>	Parameter/Data bytes
<b>Byte n:</b>	Checksum

The checksum is computed by exclusive-OR-ing every byte in the string. If the same checksum is calculated by the 18LF8722 upon receipt of the packet, then a single-byte reply of 0x55 is returned to the host. In the event of a checksum error, the 18LF8722 will return an error code of 0xAA. For example, setting port PIN\_A1 high would require the following string of bytes:

0x03, 0xA6, 0x2D, 0x01, 0x89

### Definition of the Bytes:

0x03 – Number of bytes in command (checksum excluded)

0xA6 – Command for set port pin high/low

0x29 – Affected port pin

0x01 – Desired state of port pin

0x8D – Checksum of previous 4 bytes

The port pins equate to hexadecimal numeric constants as defined below:

### PORT A:

PIN\_A0 0x2C, PIN\_A1 0x2D, PIN\_A2 0x2E,  
PIN\_A3 0x2F, PIN\_A4 0x30, PIN\_A5 0x31

### PORT B:

PIN\_B0 0x34, PIN\_B1 0x35, PIN\_B2 0x36, PIN\_B3 0x37,  
PIN\_B4 0x38, PIN\_B5 0x39, PIN\_B6 0x3A, PIN\_B7 0x3B

PORT C:

PIN\_C0 0x3C, PIN\_C1 0x3D, PIN\_C2 0x3E, PIN\_C3 0x3F,  
PIN\_C4 0x40, PIN\_C5 0x41, PIN\_C6 0x42, PIN\_C7 0x43

PORT D:

PIN\_D0 0x44, PIN\_D1 0x45, PIN\_D2 0x46, PIN\_D3 0x47,  
PIN\_D4 0x48, PIN\_D5 0x49, PIN\_D6 0x4A, PIN\_D7 0x4B

PORT E:

PIN\_E0 0x4C, PIN\_E1 0x4D, PIN\_E2 0x4E, PIN\_E3 0x4F,  
PIN\_E4 0x50, PIN\_E5 0x51, PIN\_E6 0x52, PIN\_E7 0x53

PORT F:

PIN\_F0 0x54, PIN\_F1 0x55, PIN\_F2 0x56, PIN\_F3 0x57,  
PIN\_F4 0x58, PIN\_F5 0x59, PIN\_F6 0x5A, PIN\_F7 0x5B

PORT G:

PIN\_G0 0x5C, PIN\_G1 0x5D, PIN\_G2 0x5E, PIN\_G3 0x5F,  
PIN\_G4 0x60

PORT H:

PIN\_H0 0x64, PIN\_H1 0x65, PIN\_H2 0x66, PIN\_H3 0x67,  
PIN\_H4 0x68, PIN\_H5 0x69, PIN\_H6 0x6A, PIN\_H7 0x6B

PORT J:

PIN\_J0 0x6C, PIN\_J1 0x6D, PIN\_J2 0x6E, PIN\_J3 0x6F,  
PIN\_J4 0x70, PIN\_J5 0x71, PIN\_J6 0x72, PIN\_J7 0x73

The source code for the Token I/O firmware (developed for the CCS C compiler) is available as a free download upon purchase of the DLP-245PL-G from [www.dlpdesign.com](http://www.dlpdesign.com). Example Visual C++ source code (for Windows 98/2000/XP) for communicating with the DLP-245PL-G via the Token I/O firmware is also available for download. The Windows source code also contains the port-pin definitions listed above.

## TOKEN I/O COMMAND SET

### 0xA5 – Line In – Reads the state of a single port pin

**Parameters:** Port – Select from available port pins (PIN\_A3, PIN\_B0, etc.)  
**Returns:** 1 Byte: State of the port pin (0 or 1)  
**Function:** This function will read the state of a single port pin. If the data direction for the pin selected was set to output, the direction is first set to input.  
**Example:** 0x2, 0xA5, 0x2F, 0x88 - Reads the current state of port pin PIN\_A3.

### 0xA6 – Line Out - Sets a single port pin high or low

**Parameters:** Port – Select from available port pins (PIN\_A3, PIN\_B3, etc.)  
 State – 0 or 1  
**Returns:** Undefined  
**Function:** This function will change the output state of a single port pin. If the data direction for the pin selected was set to input, the direction is first set to output.  
**Example:** 0x3, 0xA6, 0x2F, 0x1, 0x8B – Sets port pin PIN\_A3 high.

### 0xA7 – Return Board ID

**Parameters:** None  
**Returns:** 5 bytes: "245PL"  
**Function:** This function will return the boardID.  
**Example:** 0x1, 0xA7, A6 – Reads the board ID.

### 0xA8 – Setup A/D

**Parameters:** Port Configuration – Selects analog port configuration. [See command 0xA8, setup\_AD() in the example firmware for options.]  
 A/D Conversion Clock – Select the source for the A/D conversion clock. [See command 0xA8, setup\_AD() in the example firmware for options.]  
**Returns:** Undefined  
**Function:** This function will select the input channel(s) and the source for the A/D conversion clock. (Refer to the datasheet for the 18LF8722 for a detailed explanation of the conversion clock.)  
**Example:** 0x3, 0xA8, 0x00, 0x7, 0xAC – Sets all available A/D inputs on the DLP-245PL-G to analog mode (0x00) and selects Fosc/64 for 20MHz operation (0x7).

### 0xA9 – A/D Conversion

**Parameters:** Analog Port Number – Selects the analog port for the A/D conversion.  
**Returns:** 2 Bytes: The 10-bit voltage data; LSB first.  
**Function:** This function will set the channel for the A/D conversion, pause 10uS, perform the A/D conversion, and then return 2 bytes to the host (LSB first). *Command 0xA8 must have been previously called to configure the desired analog channel as an A/D input.*  
**Example:** 0x2, 0xA9, 0x2, 0xA8 – Selects analog channel 2 (AN2), performs the A/D conversion, and returns 2 bytes of data.



## 0xAA – EEPROM Read

**Parameters:** Address (2 bytes) – Selects the zero-based address of the location in the 18LF8722's internal EEPROM for reading. The first address byte sent is the MSByte, and the second is the LSByte.

**Returns:** 1 Byte: The byte of data read from the EEPROM.

**Function:** This function will read and return the selected location in the 18LF8722's internal EEPROM.

**Example:** 0x2, 0xAA, 0x1, 0x2, 0xAB – Reads EEPROM Address 258, and returns a single byte of data.

## 0xAB – EEPROM Write

**Parameters:** Address (2 bytes) – Selects the zero-based address of the location in the 18LF8722's internal EEPROM for writing. The first address byte sent is the MSByte, and the second is the LSByte.  
Data – 8-bit data to be written to the EEPROM.

**Returns:** Undefined.

**Function:** This function will write the selected location in the 18LF8722's internal EEPROM.

**Example:** 0x3, 0xAB, 0x1, 0x2, 0xFE, 0x55 – Writes a value of 0xFE to EEPROM Address 258.

## 0xAC – DS18B20 Start Convert

**Parameters:** Port Pin – Selects the microcontroller port pin on the 18LF8722 to be used for communication with a DS18B20 temperature sensor.

**Returns:** 1 Byte: A single byte indicating the result of attempting to reset the temperature sensor.

**Sensor Status Definitions:**

- 99: Sensor ready
- 8: Short circuit detected
- 2: No sensor detected

**Function:** This function will select a port pin for communication with a DS18B20 temperature sensor and start a temperature conversion. In addition to power and ground connections, a 1.5K to 4.7K pull-up resistor must be connected between VCC and the data pin of the DS18B20 for proper communication. The conversion time for the DS18B20 is specified for a maximum of 750 milliseconds.

**Example:** 0x2, 0xAC, 0x34, 0x9A – Starts a temperature conversion on 18LF8722 Port Pin B0.

## 0xAD – DS18B20 Read Data

**Parameters:** Port Pin – Selects the microcontroller port pin on the 18LF8722 to be used for communication with the DS18B20 temperature sensor.

**Returns:** 9 Bytes: The contents of the scratchpad memory in the DS18B20.

**Function:** This function will select a port pin for communication with a DS18B20 temperature sensor and request the current contents of the DS18B20's scratchpad memory. A conversion (command 0xAC) must have been previously requested for the data to be current. Additionally, host software must wait at least 750 milliseconds for the conversion to complete before reading the 9 bytes of data. In addition to power and ground, a 1.5K to 4.7K pull-up resistor must be connected between VCC and the data pin of the DS18B20 for proper



communications. (For an example of how to calculate .1°C resolution temperature data from the 9 bytes of return data, refer to the example Windows source code.)

**Example:** 0x2, 0xAD, 0x34, 0x9B – Requests 9-byte temperature data from 18LF8720 Port Pin B0.

## 0xAF – Loopback

**Parameters:** Data byte – The byte of data to be looped back to the host.

**Returns:** 1 Byte: The data byte written.

**Function:** This function will echo the specified byte of data back to the host. The port pins are not affected.

**Example:** 0x2, 0xAF, 0x12, 0xBF – Requests the data byte 0x12 to be echoed back to the host.

**0x55 - Port A Read**

**0x59 - Port C Read**

**0x5B - Port D Read**

**0x5D - Port E Read**

**0x65 - Port F Read**

**0x69 - Port H Read**

**0x6B - Port J Read**

**Parameters:** None

**Returns:** 1 Byte: The 8-bit data read from the selected port.

**Function:** This function will read the specified 8-bit port and return the results to the host. The port is automatically configured as an input port. Analog input ports must be configured for digital I/O (command 0xA8) prior to using this command.

**Example:** 0x1, 0x5B, 0x5A – Requests a read from the 8-bit data bus Port D.

**0x56 - Port A Write**

**0x5A - Port C Write**

**0x5C - Port D Write**

**0x5E - Port E Write**

**0x66 - Port F Write**

**0x6A - Port H Write**

**0x6C - Port J Write**

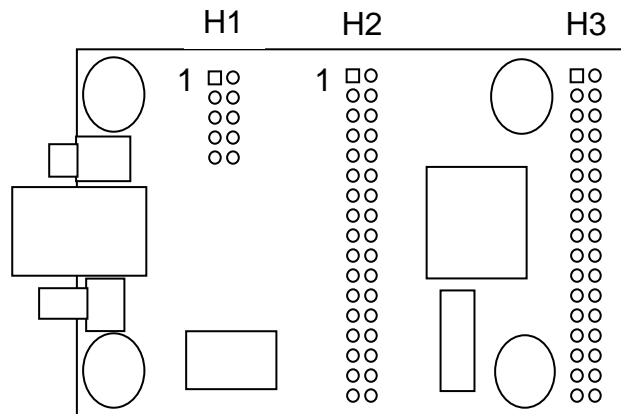
**Parameters:** Data byte – The byte of data to be written to the 8-bit port.

**Returns:** Undefined

**Function:** This function will write the specified data byte to the 8-bit port. Data is latched on the port pins until changed by another command. The port is automatically configured as an output port. Port A must be configured for digital I/O (command 0xA8) prior to using this command.

**Example:** 0x2, 0x5C, 0x12, 0x4C – Writes the data byte 0x12 to the 8-bit data bus Port D.

**TABLE 1: DLP-245PL PINOUT DESCRIPTION**



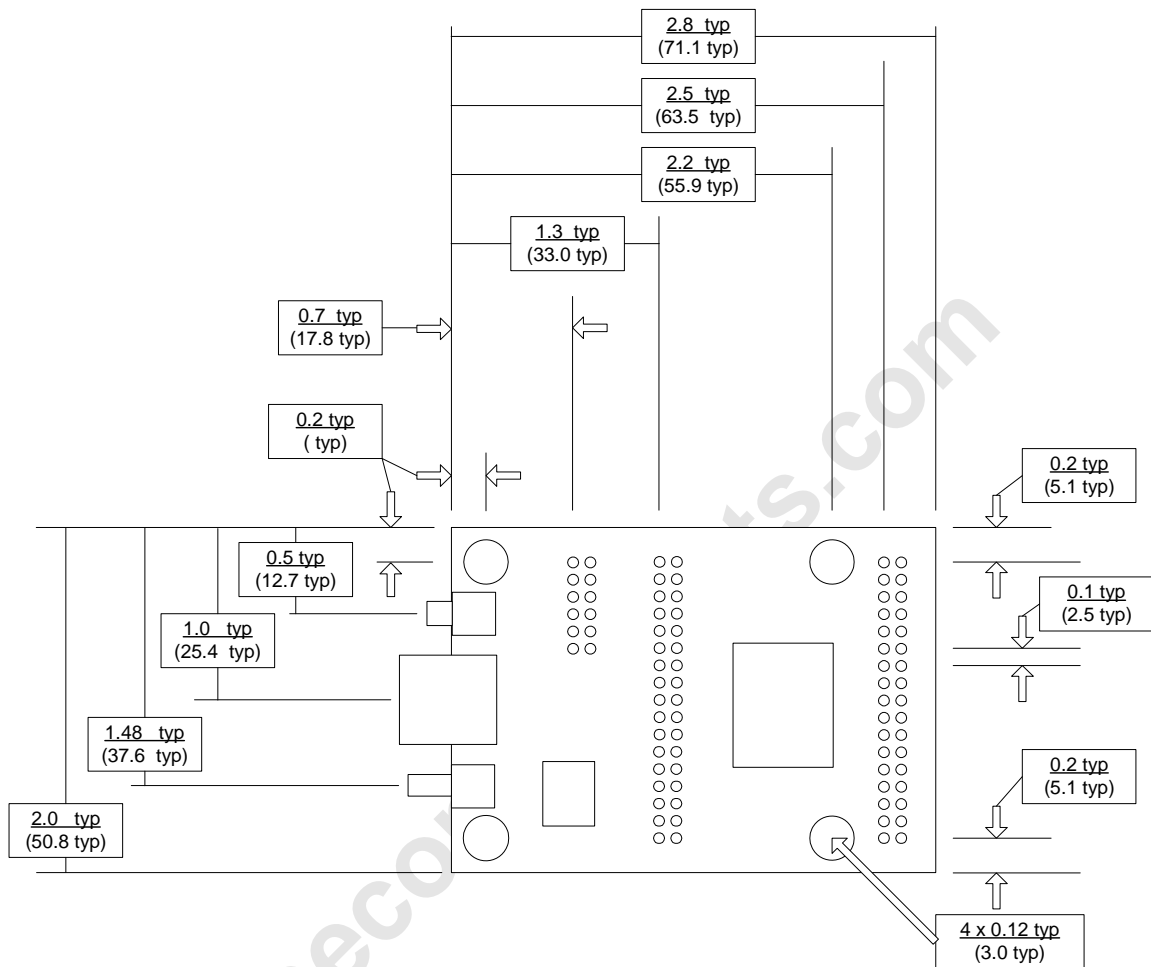
Pin #	H1 Connector Description
1	<b>EXT5V Input</b> 5-volt power input for board if not using USB port power.
2	<b>GROUND</b>
3	<b>EXT5V Input</b> 5-volt power input for board if not using USB port power.
4	<b>MCLR</b> Pull to ground to reset the microcontroller.
5	<b>GROUND</b>
6	<b>GROUND</b>
7	<b>EXT3-5V</b> Power input for VCCIO pin on the USB chip and microcontroller. Lower voltages require running the microcontroller at 6MHz (PLL turned off in configuration bits).
8	<b>GROUND</b>
9	<b>EXT3-5V</b> Power input for VCCIO pin on the USB chip and microcontroller. Lower voltages require running the microcontroller at 6MHz (PLL turned off in configuration bits).
10	<b>GROUND</b>

	H2 Connector Description
1	<b>F6</b> (I/O) Port Pin F6 connected to the 18LF8722 microcontroller. Analog input Channel 11.
2	<b>F7</b> (I/O) Port Pin F7 connected to the 18LF8722 microcontroller.
3	<b>F4</b> (I/O) Port Pin F4 connected to the 18LF8722 microcontroller. Analog input Channel 9.
4	<b>F5</b> (I/O) Port Pin F5 connected to the 18LF8722 microcontroller. Analog input Channel 10.
5	<b>F2</b> (I/O) Port Pin F2 connected to the 18LF8722 microcontroller. Analog input Channel 7.
6	<b>F3</b> (I/O) Port Pin F3 connected to the 18LF8722 microcontroller. Analog input Channel 8.
7	<b>H6</b> (I/O) Port Pin H6 connected to the 18LF8722 microcontroller. Analog input Channel 14.
8	<b>H7</b> (I/O) Port Pin H7 connected to the 18LF8722 microcontroller. Analog input Channel 15.
9	<b>H4</b> (I/O) Port Pin H4 connected to the 18LF8722 microcontroller. Analog input Channel 12.
10	<b>H5</b> (I/O) Port Pin H5 connected to the 18LF8722 microcontroller. Analog input Channel 13.
11	<b>F0</b> (I/O) Port Pin F0 connected to the 18LF8722 microcontroller. Analog input Channel 5.
12	<b>F1</b> (I/O) Port Pin F1 connected to the 18LF8722 microcontroller. Analog input Channel 6.
13	<b>A4</b> (I/O) Port Pin A4 connected to the 18LF8722 microcontroller. Open drain output.
14	<b>A5</b> (I/O) Port Pin A5 connected to the 18LF8722 microcontroller. Analog input Channel 4.
15	<b>A2</b> (I/O) Port Pin A2 connected to the 18LF8722 microcontroller. Analog input Channel 2.
16	<b>A3</b> (I/O) Port Pin A3 connected to the 18LF8722 microcontroller. Analog input Channel 3.
17	<b>A0</b> (I/O) Port Pin A0 connected to the 18LF8722 microcontroller. Analog input Channel 0.
18	<b>A1</b> (I/O) Port Pin A1 connected to the 18LF8722 microcontroller. Analog input Channel 1.
19	<b>GROUND</b>

20	<b>GROUND</b>
21	<b>GROUND</b>
22	<b>GROUND</b>
23	<b>H0</b> (I/O) Port Pin H0 connected to the 18LF8722 microcontroller.
24	<b>H1</b> (I/O) Port Pin H1 connected to the 18LF8722 microcontroller.
25	<b>H2</b> (I/O) Port Pin H2 connected to the 18LF8722 microcontroller.
26	<b>H3</b> (I/O) Port Pin H3 connected to the 18LF8722 microcontroller.
27	<b>DB0</b> (I/O) Line 0 of the data bus between the 18LF8722 and the FT245RL USB-FIFO.
28	<b>DB1</b> (I/O) Line 1 of the data bus between the 18LF8722 and the FT245RL USB-FIFO.
29	<b>DB2</b> (I/O) Line 2 of the data bus between the 18LF8722 and the FT245RL USB-FIFO.
30	<b>DB3</b> (I/O) Line 3 of the data bus between the 18LF8722 and the FT245RL USB-FIFO.
31	<b>DB4</b> (I/O) Line 4 of the data bus between the 18LF8722 and the FT245RL USB-FIFO.
32	<b>DB5</b> (I/O) Line 5 of the data bus between the 18LF8722 and the FT245RL USB-FIFO.
33	<b>DB6</b> (I/O) Line 6 of the data bus between the 18LF8722 and the FT245RL USB-FIFO.
34	<b>DB7</b> (I/O) Line 7 of the data bus between the 18LF8722 and the FT245RL USB-FIFO.

	<b>H3 Connector Description</b>
1	<b>USBVCC</b> 5V power from the USB port.
2	<b>E0</b> (I/O) Port Pin E0 connected to the 18LF8722 microcontroller.
3	<b>SWVCC</b>
4	<b>E1</b> (I/O) Port Pin E1 connected to the 18LF8722 microcontroller.
5	<b>SWVCC</b>
6	<b>E2</b> (I/O) Port Pin E2 connected to the 18LF8722 microcontroller.
7	<b>G0</b> (I/O) Port Pin G0 connected to the 18LF8722 microcontroller.
8	<b>E3</b> (I/O) Port Pin E3 connected to the 18LF8722 microcontroller.
9	<b>G3</b> (I/O) Port Pin G3 connected to the 18LF8722 microcontroller.
10	<b>E4</b> (I/O) Port Pin E4 connected to the 18LF8722 microcontroller.
11	<b>B0</b> (I/O) Port Pin B0 connected to the 18LF8722 microcontroller.
12	<b>E5</b> (I/O) Port Pin E5 connected to the 18LF8722 microcontroller.
13	<b>B1</b> (I/O) Port Pin B1 connected to the 18LF8722 microcontroller.
14	<b>E6</b> (I/O) Port Pin E6 connected to the 18LF8722 microcontroller.
15	<b>B3</b> (I/O) Port Pin B3 connected to the 18LF8722 microcontroller.
16	<b>E7</b> (I/O) Port Pin E7 connected to the 18LF8722 microcontroller.
17	<b>GROUND</b>
18	<b>GROUND</b>
19	<b>J0</b> (I/O) Port Pin J0 connected to the 18LF8722 microcontroller.
20	<b>C0</b> (I/O) Port Pin C0 connected to the 18LF8722 microcontroller.
21	<b>J1</b> (I/O) Port Pin J1 connected to the 18LF8722 microcontroller.
22	<b>C1</b> (I/O) Port Pin C1 connected to the 18LF8722 microcontroller.
23	<b>J2</b> (I/O) Port Pin J2 connected to the 18LF8722 microcontroller.
24	<b>C2</b> (I/O) Port Pin C2 connected to the 18LF8722 microcontroller.
25	<b>J3</b> (I/O) Port Pin J3 connected to the 18LF8722 microcontroller.
26	<b>C3</b> (I/O) Port Pin C3 connected to the 18LF8722 microcontroller.
27	<b>J4</b> (I/O) Port Pin J4 connected to the 18LF8722 microcontroller.
28	<b>C4</b> (I/O) Port Pin C4 connected to the 18LF8722 microcontroller.
29	<b>LED</b> (I/O) Port Pin J5 connected to the 18LF8722 microcontroller. LED controlled by 18LF8722.
30	<b>C5</b> (I/O) Port Pin C5 connected to the 18LF8722 microcontroller.
31	<b>J6</b> (I/O) Port Pin J6 connected to the 18LF8722 microcontroller.
32	<b>C6</b> (I/O) Port Pin C6 connected to the 18LF8722 microcontroller.
33	<b>J7</b> (I/O) Port Pin J7 connected to the 18LF8722 microcontroller.
34	<b>C7</b> (I/O) Port Pin C7 connected to the 18LF8722 microcontroller.

**MECHANICAL DRAWINGS (PRELIMINARY)**  
**INCHES (MILLIMETERS) UNLESS OTHERWISE NOTED**



## DISCLAIMER

Neither the whole nor any part of the information contained within or the product described in this datasheet may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder.

This product and its documentation are supplied on an as-is basis, and no warranty as to their suitability for any particular purpose is either made or implied. DLP Design will not accept any claim for damages whatsoever arising as a result of use or failure of this product. Your statutory rights are not affected.

This product or any variant of it is not intended for use in any medical appliance, device, or system in which the failure of the product might reasonably be expected to result in personal injury.

This document provides preliminary information that may be subject to change without notice.

## CONTACT INFORMATION

DLP Design, Inc.  
1605 Roma Ln.  
Allen, TX 75013

Phone: 469-964-8027  
Fax: 415-901-4859  
Email: [support@dlpdesign.com](mailto:support@dlpdesign.com)  
Internet: <http://www.dlpdesign.com>

